Operations Research, Spring 2024 (112-2)
Homework 2

Ling-Chieh Kung*

March 25, 2023

1 Rules

- This homework is due at ***23:59, April 13***. Submissions late by no more than twelve hours get 10 points off; those late by no more than 24 hours get 20 points off; those late by more than 24 hours get no point.
- For this homework, students should work individually. While discussions are encouraged, copying is prohibited.
- Please submit a ***PDF file*** through NTU COOL and make sure that the submitted work contains the student ID and name. Those who fail to do these will get 10 points off.
- You are required to ***type*** your work with LaTeX (strongly suggested) or a text processor with a formula editor. Hand-written works are not accepted. You are responsible to make your work professional in mathematical writing by following at least the following rules:[1]
  1. When there is a symbol denoted by an English letter, make it italic. For example, write $a + b = 3$ rather than a + b = 3.
  2. An operator (e.g., +) should not be italic. A function with a well-known name (e.g., log, max and sin) is considered as an operator.
  3. A number should not be italic. For example, it should be $a + b = 3$ rather than $a + b = 3$.
  4. Superscripts or subscripts should be put in the right positions. For example, a_1 and $a1$ are completely different: The former is a variable called a_1 while the latter is actually $a \times 1$.
  5. When there is a subtraction, write $-$ rather than -. For example, write $a - b = 3$ rather than a - $b = 3$. The same thing applies to the negation operator. For example, write $a = -3$ rather than a = -3.
  6. If you want to write down the multiplication operator, write $\times$ rather than *.
  7. For an exponent, write it as a superscript rather than using ^. For example, write 10^2 rather than 10^2.
  8. There should be proper space beside a binary operator. For example, it should be $a + b = 3$ rather than a+b=3.

  Those who fail to follow these rules may get at most 10 points off.
- As we may see, there are many students, many problems, but only a few TAs. Therefore, when the TAs grade this homework, it is possible for only some problems to be randomly selected and graded. For all problems, detailed suggested solutions will be provided.

*Department of Information Management, National Taiwan University. E-mail: lckung@ntu.edu.tw.

[1] A more complete list of formatting rules is on NTU COOL.

2 Problems

1. (20 points; 5 points each) Consider the following LP

$$\begin{aligned} \min \quad & x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 4 \\ & x_1 + 2x_2 \leq 9 \\ & x_2 \geq 3 \\ & x_i \geq 0 \quad \forall i = 1, 2. \end{aligned}$$

   (a) Find the standard form of this LP.

   (b) List all the basic solutions and basic feasible solutions for the standard form of this LP.

   (c) List all the extreme points of the feasible region of the original LP. DO NOT prove that they are extreme points; just list them. Then show that there is a one-to-one correspondence between basic feasible solutions and extreme points.

   (d) Use the simplex method (with the two-phase implementation, if needed) and the smallest index rule to solve the LP. Write down the complete process, optimal solution to the original LP, and its associated objective value. Is there any iteration that has no improvement?

2. (10 points; 5 points each) Consider the integer program

$$\begin{aligned} \max \quad & 2x_1 + 2x_2 + 5x_3 + 11x_4 + 10x_5 + 3x_6 - 6x_7 \\ \text{s.t.} \quad & x_1 + 4x_2 + 3x_3 + 5x_4 + 3x_5 - 4x_6 + 2x_7 \leq 6 \\ & x_i \in \{0, 1\} \quad \forall i = 1, ..., 7, \end{aligned}$$

   which is a variant of the knapsack problem.

   (a) Use the greedy algorithm introduced in class to solve the linear relaxation of this integer program.

   (b) Use the branch-and-bound algorithm to solve the original integer program. Depict the full branch-and-bound tree. Do not write down the solution process of each node; write down just an optimal solution and its objective value of each node.

3. (15 points) There are m towns in a county. The county government is considering where to build at least p landfills in n potential locations. The distance between town i and location j is d_{ij}. The population at town i is h_i. The government wants to maximize the average distances between each person and her/his closest landfill.

   (a) (5 points) Formulate an integer program to achieve this goal. Use your own words to explain the formulation in details.

   Hint. This problem has appeared in one lecture problem.

   Note. If you believe ChatGPT may help you explain the formulation, please feel free to use it. All you need to do is to make sure that the explanation is correct, precise, and concise.

   (b) (10 points) Consider the two instances contained in the file "OR112-2_hw02_data.xlsx" (if you do not use Microsoft Excel, you may use Google Spreadsheet to open the file). In each sheet, which contains an instance, parameter symbols are in blue cells, indices are in orange cells, and parameter values are in green cells. For example, in instance 2, $m = 20$, $n = 10$, $p = 5$, $h_2 = 28$, and $d_{12,5} = 164$.

   Write Python to invoke Gurobi Optimizer to solve the two instances. Submit your Python program by copying and pasting it into your answer. You should not hard code those parameter values; please put the parameter values in a file and read data from the file to retrieve the values. There is no need to explain the program in your answer; just submit it. Moreover, report the optimal solutions (and their objective values) you find for the two instances.

If you prefer, you may write your program in other languages; if you prefer, you may use other optimization library (as long as you ensure that it generates a correct optimal solution). Nevertheless, please understand that while we provide instruction on using Python to invoke Gurobi Optimizer, there is nothing we may help with other combinations.

4. (35 points) Continue from Problem 3.

   (a) (10 points) Consider the following greedy algorithm designed for the landfill location problem. The algorithm runs in $n - p$ iterations. Initially, the algorithm starts with a feasible solution that builds a landfill in each candidate locations. In each iteration, it examines all landfills to see if one and exactly one landfill can be removed, removing which one results in the maximum objective value in that iteration (and if there is a tie, it chooses the landfill with the smallest index). It keeps removing landfills until only p landfills remain.

   Implement this greedy algorithm in Python or any language you like. Submit your program by copying and pasting it into your answer. You should not hard code those parameter values; please put the parameter values in a file and read data from the file to retrieve the values. There is no need to explain the program in your answer; just submit it. Moreover, report the solutions (and their objective values) you find for the two instances.

   (b) (10 points) Consider another heuristic algorithm designed for the landfill location problem based on linear relaxation. The idea is simple. Given an instance, first we solve its linear relaxation to obtain a probably fractional LP-optimal solution x^{LP}. We then pick the largest p values and set the corresponding x_j^{LP} to 1 (if there is a tie, pick those with smaller indices); the remaining $n - p$ variables are set to 0.

   Implement this heuristic algorithm in Python or any language you like. When solving the linear relaxation, use Gurobi Optimizer or any optimization library you like. Submit your program by copying and pasting it into your answer. You should not hard code those parameter values; please put the parameter values in a file and read data from the file to retrieve the values. There is no need to explain the program in your answer; just submit it. Moreover, report the solutions (and their objective values) you find for the two instances.

   (c) (5 points) Let z_k^G be the objective value of the solution found by the greedy algorithm in Part (a) for instance $k \in \{1, 2\}$. Similarly, let z_k^R be that by the heuristic algorithm in Part (b) for instance $k \in \{1, 2\}$. For each of the two instances, report z_k^{IP}, z_k^{LP}, z_k^G, z_k^R, and the four percentage optimality gaps (each algorithm has two optimality gaps, one uses z_k^{IP} and one uses z_k^{LP}). In average, which algorithm performs better in these two instances?

   (d) (10 points) Comment on the time complexity (with the big-O notation) and performance of the two heuristic algorithms. If you need to solve the landfill location problem with $m = 500$, $n = 100$, and $p = 20$, which algorithm do you prefer? Why?

   Hint. The complexity of using the simplex method to solve a linear program with m constraints in roughly $O(m^3)$.[2] Though the way Gurobi Optimizer solves a linear program is more complicated than just running the classic simplex method, the complexity is pretty much at the same order. To answer this problem, please use the $O(m^3)$ fact directly.

5. (20 points; 10 points each) Consider the following nonlinear program

$$\min \ 3x_1^2 + 2x_2^2 + 4x_1x_2 + 6e^{x_1} + x_2.$$

   Later when needed, use numerical solutions rather than analytical solutions. For example, when solving $-x = e^x$, using any calculator or software to find $x = -0.567$ as a numerical solution is good enough. There is no need to analytically solve $-x = e^x$.

   (a) Start from $(x_1, x_2) = (0, 0)$ to run one iteration of the gradient descent method to solve this instance. In this iteration, move to the global minimum along the direction you choose. Write down the detailed process of the iteration.

   (b) Start from $(x_1, x_2) = (0, 0)$ to run one iteration of the Newton's method to solve this instance. Write down the detailed process of the iteration.

[2] While this will be briefly mentioned in the third module of this semester, the instructor forgot that this has not been mentioned in module 2.